Iterative Soft-Decision Decoding of Reed-Solomon Convolutional Concatenated Codes

Li Chen and Xiao Ma
School of Information Science and Technology, Sun Yat-sen University,
Guangzhou, China. Email: {chenli55, maxiao}@mail.sysu.edu.cn

***Abstract*—Reed-Solomon convolutional concatenated (RSCC) code has been widely applied in wireless and space communications. However, iterative soft-decision decoding of the concatenated code is yet to be developed. This paper proposes a novel iterative soft decoding algorithm for the concatenated coding scheme. The maximum *a posteriori* (MAP) algorithm is used to decode the inner convolutional code. Its soft output will be deinterleaved and then passed to the soft-in-soft-out (SISO) decoding algorithm for the outer Reed-Solomon (RS) code. The outer SISO decoder integrates the adaptive belief propagation (ABP) algorithm and the Koetter-Vardy (KV) list decoding algorithm, attempting to find out the transmitted message. If it is found, the deterministic probabilities of the corresponding RS coded bits will be fed back. Otherwise, the extrinsic probabilities that are yielded by the ABP algorithm will be given as the feedback. With the proposed soft information exchange decoding mechanism, error-correction potential of the concatenated code can be better exploited. Our simulation results show that significant performance improvement can be achieved over the existing decoding algorithms.**

***Index Terms*—Concatenated codes, convolutional codes, iterative decoding, Reed-Solomon codes, soft-decision decoding**

I. INTRODUCTION

Concatenated codes were first introduced by Forney [1]. It has been shown that concatenating a nonbinary outer code and a binary inner code could constitute a capacity approaching error-correction code. The legacy Reed-Solomon convolutional concatenated (RSCC) code is an example. The inner convolutional code is good at correcting spread bit errors, while the outer Reed-Solomon (RS) code is good at correcting burst errors. Such combinatorial functions ensure the RSCC codes' strong error-correction capability and their application can be widely found in wireless and space communications [2] [3].

The conventional decoding scheme for RSCC codes employs the Viterbi algorithm and the Berlekamp-Massey (BM) algorithm to decode the inner and outer codes, respectively. A block interleaver (and deinterleaver) is employed between the inner and outer codes in order to spread the burst errors resulted from the Viterbi decoding. The primitive attempt to decode the RSCC codes iteratively was proposed in [4]. However, the BM algorithm was used to decode the outer code, preventing the soft information being given as the feedback. Another attempt to improve the concatenated code's error-correction performance is to employ a stronger RS decoding algorithm, e.g., the Koetter-Vardy (KV) algorithm [5]. Utilizing the KV algorithm for the outer code in the iterative decoding mechanism of [4] was considered in [6]. Finally, collaborative decoding of the interleaved RS codes has also been considered for the concatenated codes [7]. Decoding the interleaved RS codewords jointly allows the BM algorithm to correct symbol errors beyond the half distance bound.

The development of turbo codes [8] showed that allowing two decoders to exchange soft information iteratively can enable a concatenated code to achieve a capacity approaching performance. However, a truly iterative soft decoding algorithm for RSCC codes is yet to be developed. This is due to the challenge in designing an efficient soft-in-soft-out (SISO) decoding algorithm for the outer code. Addressing the problem, this paper proposes an iterative soft-decision decoding algorithm for the RSCC codes. The maximum *a posteriori* (MAP) [9] algorithm is used to decode the inner code, delivering the extrinsic probabilities for the interleaved RS coded bits. They are then deinterleaved and mapped to the *a priori* probabilities of the RS coded bits, which will be utilized by the outer SISO decoder. The outer SISO decoder has two successive stages. The first stage performs the belief propagation (BP) decoding based on the adapted binary parity-check matrix of the RS code, namely the adaptive BP (ABP) [10] [11]. It delivers both the extrinsic and *a posteriori* probabilities of the RS coded bits. Utilizing the *a posteriori* probabilities, the second stage, i.e., the KV algorithm, is performed to retrieve the transmitted message. If the message is found, the deterministic probabilities of the corresponding RS coded bits will be fed back. Otherwise, the extrinsic probabilities will be fed back. They are then interleaved and mapped to the *a priori* probabilities of the interleaved RS coded bits for the next round MAP decoding. The proposed decoding algorithm allows the extrinsic probabilities of the RS coded bits to be iterated between the SISO decoders of the inner and outer codes efficiently. Consequently, it can well exploit the error-correction potential of the RSCC codes, and our simulation results show a significant performance improvement over the existing decoding algorithms.

II. THE RSCC CODES

Let $\mathbb{F}_q = \{\rho_1, \rho_2, \ldots, \rho_q\}$ denote the finite field of size q. It is assumed to be an extension field of $\mathbb{F}_2$ and $q = 2^\omega$, where ω is a positive integer. Let $\mathbb{F}_q[x]$ and $\mathbb{F}_q[x, y]$ denote the rings of univariate and bivariate polynomials defined over $\mathbb{F}_q$, respectively. The encoder of RSCC codes is shown by Fig.1. There is a block interleaver between the inner and outer

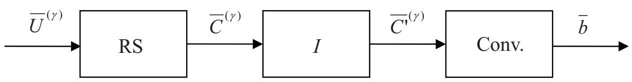

Fig. 1. Block diagram of the RSCC encoder.

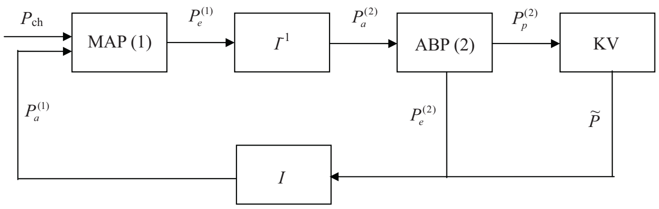

Fig. 2. Block diagram of the iterative soft-decision decoding algorithm.

encoders. Let D denote the depth of the block interleaver and γ denote the index of the RS codeword where $1 \leq \gamma \leq D$.

The message vector of an (n, k) RS code can be written as

$$\overline{U}^{(\gamma)} = [U_1^{(\gamma)}\ U_2^{(\gamma)}\ \ldots\ U_k^{(\gamma)}] \in \mathbb{F}_q^k, \tag{1}$$

where n and k are the length and dimension of the code, respectively. The superscript (γ) denotes the variable belongs to the γth RS codeword. Its generator matrix $\mathbf{G}$ is

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \ldots & \alpha^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{k-1} & \ldots & \alpha^{(k-1)(n-1)} \end{pmatrix}, \tag{2}$$

where α is a primitive element of $\mathbb{F}_q$. The γth RS codeword can be generated by

$$\overline{C}^{(\gamma)} = \overline{U}^{(\gamma)} \cdot \mathbf{G} = [C_1^{(\gamma)}\ C_2^{(\gamma)}\ \ldots\ C_n^{(\gamma)}] \in \mathbb{F}_q^n. \tag{3}$$

For the (n, k) RS code, its parity-check matrix $\mathbf{H}$ is

$$\mathbf{H} = \begin{pmatrix} 1 & \alpha & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \ldots & \alpha^{2(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{n-k} & \ldots & \alpha^{(n-k)(n-1)} \end{pmatrix}. \tag{4}$$

Let $\sigma(x) \in \mathbb{F}_2[x]$ denote a primitive polynomial of $\mathbb{F}_q$ and $\mathbf{A}$ be its companion matrix with size $\omega \times \omega$. The binary parity-check matrix $\mathbf{H}_b$ of the RS code can be generated by mapping the entries of $\mathbf{H}$ as $\alpha^i \mapsto \mathbf{A}^i$, where $i = 0, 1, \ldots, q-2$.

After the D RS codewords have been generated, they will be interleaved according to the interleaving function I which has a vertical read-in and horizontal read-out interleaving pattern. The γth interleaved RS codeword is

$$\overline{C'}^{(\gamma)} = [C_1'^{(\gamma)}\ C_2'^{(\gamma)}\ \ldots\ C_n'^{(\gamma)}] \in \mathbb{F}_q^n. \tag{5}$$

Note that the interleaved codeword vector $\overline{C'}^{(\gamma)}$ may not be a valid RS codeword. All the interleaved RS codewords $\overline{C'}^{(1)}, \overline{C'}^{(2)}, \ldots, \overline{C'}^{(D)}$ will then be converted into a binary interleaved coded bit sequence

$$\begin{aligned} &c_1', c_2', \ldots, c_{n\omega}', c_{n\omega+1}', c_{n\omega+2}', \ldots, c_{2n\omega}', \ldots, \\ &c_{(D-1)n\omega+1}', c_{(D-1)n\omega+2}', \ldots, c_{Dn\omega}'. \end{aligned} \tag{6}$$

They are the input to the inner encoder. In this paper, the rate half non-systematically encoded convolutional (NSC) code is used as the inner code. The transfer functions $G_i(x) \in \mathbb{F}_2[x]$ of the inner code can be written as

$$G_i(x) = \sum_{j=0}^{\varpi} g_j^{(i)} x^j, \tag{7}$$

where $i = 1, 2$ and ϖ denotes the number of shift registers in the encoder and the code's constraint length is $\varpi + 1$. Note that ϖ zero padding bits $c_{Dn\omega+1}', \ldots, c_{Dn\omega+\varpi}'$ will be appended to the end of the bit sequence (6) to drive the encoder back to the all-zero state. With the above mentioned input, the convolutional codeword is

$$\bar{b} = [b_1\ b_2\ \cdots\ b_N] \in \mathbb{F}_2^N, \tag{8}$$

where $N = 2(Dn\omega + \varpi)$.

III. Iterative Soft-Decision Decoding

The block diagram of the proposed algorithm is shown by Fig.2. We use P_a, P_e and P_p to denote the *a priori* probability, the extrinsic probability and the *a posteriori* probability, respectively. Superscripts (1) and (2) are used to indicate the probabilities are associated with the MAP algorithm and the ABP algorithm, respectively. Moreover, P_{ch} denotes the channel observations which will be left unchanged during the decoding. $\tilde{P}$ denotes the deterministic probability that is estimated by the KV decoding and $\tilde{P} \in \{0, 1\}$.

With the channel observations P_{ch} and the *a priori* probabilities $P_a^{(1)}$ of the interleaved RS coded bits, the MAP algorithm is performed to determine their extrinsic probabilities $P_e^{(1)}$. They are then deinterleaved and mapped to the *a priori* probabilities $P_a^{(2)}$ of the RS coded bits. For each RS codeword, the ABP algorithm is performed, delivering the extrinsic probabilities $P_e^{(2)}$ and the *a posteriori* probabilities $P_p^{(2)}$ of the RS coded bits. With $P_p^{(2)}$, the KV algorithm is performed to find the transmitted RS codeword. Once it is found, the deterministic probabilities $\tilde{P}$ are yielded. Afterwards, both $\tilde{P}$ of the decoded bits and $P_e^{(2)}$ of the undecoded bits are interleaved and mapped back to the *a priori* probabilities $P_a^{(1)}$ for the next round MAP decoding. The decoding terminates when either all the RS codewords have been decoded or the maximal iteration number N_{ITER} is reached.

A. SISO Decoding of the Inner Code

For the sake of consistency, we will describe the inner SISO decoding in the light of the rate half NSC code. Let $\mathcal{S} = \{1, 2, \ldots, \Omega\}$ denote the set of states of the inner code's trellis and $|\mathcal{S}| = \Omega = 2^{\varpi}$, and let $\theta \in \{0, 1\}$. Given $\overline{\mathcal{Y}} \in \mathbb{R}$ as the received vector observed from the channel, the channel observation can be obtained

$$P_{\text{ch},j}(\theta) = \Pr[b_j = \theta \mid \overline{\mathcal{Y}}], \tag{9}$$

where $j = 1, 2, \ldots, N$. The *a priori* probabilities of the interleaved RS coded bits c_j' are defined as

$$P_{a,j}^{(1)}(\theta) = \Pr[c_j' = \theta], \tag{10}$$

where $j = 1, 2, \ldots, Dn\omega + \varpi$. At the beginning of the decoding, they are initialized as $P_{a,j}^{(1)}(0) = P_{a,j}^{(1)}(1) = 0.5$ for $j = 1, 2, \ldots, Dn\omega$, and $P_{a,j}^{(1)}(0) = 1$ and $P_{a,j}^{(1)}(1) = 0$ for $j = Dn\omega + 1, \ldots, Dn\omega + \varpi$. Let us assume that at the time instant j, an input of $c'_j = \theta$ corresponds to two coded bits of $b_{2j-1}b_{2j} = \theta_1\theta_2$, where $(\theta_1, \theta_2) \in \{0, 1\}$. It triggers a trellis state transition from $\mathcal{X}_j$ to $\mathcal{X}_{j+1}$, where $(\mathcal{X}_j, \mathcal{X}_{j+1}) \in \mathcal{S}$. The state transition probability is determined by

$$\Gamma_j(\mathcal{X}_j, \mathcal{X}_{j+1}) = P_{a,j}^{(1)}(\theta) \cdot P_{\text{ch},2j-1}(\theta_1) \cdot P_{\text{ch},2j}(\theta_2). \quad (11)$$

The MAP algorithm [9] will then perform the forward and the backward traces to determine the *a posteriori* probabilities of the interleaved RS coded bits c'_j, which are defined as

$$P_{p,j}^{(1)}(\theta) = \Pr[c'_j = \theta \mid \overline{\mathcal{Y}}]. \quad (12)$$

Let $\mathcal{A}_j(\mathcal{X}_j)$ denote the probability of the trellis ends at state $\mathcal{X}_j$ at the time instant j. They are initialized as $\mathcal{A}_1(1) = 1$ and $\mathcal{A}_1(2) = \cdots = \mathcal{A}_1(\Omega) = 0$ [1]. The forward trace determines the probability of the trellis ends at state $\mathcal{X}_{j+1}$ at the time instant $j + 1$ by

$$\mathcal{A}_{j+1}(\mathcal{X}_{j+1}) = \mathcal{N}_{\mathcal{A}} \sum_{\mathcal{X}_j=1}^{\Omega} \mathcal{A}_j(\mathcal{X}_j)\Gamma_j(\mathcal{X}_j, \mathcal{X}_{j+1}), \quad (13)$$

where $\mathcal{N}_{\mathcal{A}} = (\sum_{\mathcal{X}_{j+1}=1}^{\Omega} \mathcal{A}_{j+1}(\mathcal{X}_{j+1}))^{-1}$. Similarly, by knowing the probability of the trellis ends at state $\mathcal{X}_{j+1}$ at the time instant $j+1$, the backward trace determines the probability of the trellis ends at state $\mathcal{X}_j$ at the time instant j by

$$\mathcal{B}_j(\mathcal{X}_j) = \mathcal{N}_{\mathcal{B}} \sum_{\mathcal{X}_{j+1}=1}^{\Omega} \mathcal{B}_{j+1}(\mathcal{X}_{j+1})\Gamma_j(\mathcal{X}_j, \mathcal{X}_{j+1}), \quad (14)$$

where $\mathcal{N}_{\mathcal{B}} = (\sum_{\mathcal{X}_j=1}^{\Omega} \mathcal{B}_j(\mathcal{X}_j))^{-1}$. Again, initializations of $\mathcal{B}_{Dn\omega+\varpi+1}(1) = 1$ and $\mathcal{B}_{Dn\omega+\varpi+1}(2) = \cdots = \mathcal{B}_{Dn\omega+\varpi+1}(\Omega) = 0$ are made. Let T_θ denote the set of state transitions that are triggered by an input of $c'_j = \theta$ as

$$T_\theta = \{\mathcal{X}_j \to \mathcal{X}_{j+1} \mid c'_j = \theta,\ (\mathcal{X}_j, \mathcal{X}_{j+1}) \in \mathcal{S}\}. \quad (15)$$

After the forward and backward traces, the *a posteriori* probabilities of the interleaved RS coded bits c'_j are determined by

$$P_{p,j}^{(1)}(\theta) = \mathcal{N}_P \sum_{(\mathcal{X}_j, \mathcal{X}_{j+1}) \in T_\theta} \mathcal{A}_j(\mathcal{X}_j)\Gamma_j(\mathcal{X}_j, \mathcal{X}_{j+1})\mathcal{B}_{j+1}(\mathcal{X}_{j+1}), \quad (16)$$

where $\mathcal{N}_P = (\sum_{\theta\in\{0,1\}} P_{p,j}^{(1)}(\theta))^{-1}$. The extrinsic probabilities of the interleaved RS coded bits c'_j can be determined by

$$P_{e,j}^{(1)}(\theta) = \mathcal{N}_E \frac{P_{p,j}^{(1)}(\theta)}{P_{a,j}^{(1)}(\theta)} \quad (17)$$

for $j = 1, 2, \ldots, Dn\omega$, and $\mathcal{N}_E = (\sum_{\theta\in\{0,1\}} P_{e,j}^{(1)}(\theta))^{-1}$.

The extrinsic probabilities $P_{e,j}^{(1)}(\theta)$ are then deinterleaved according to the deinterleaving function I^{-1}. Notice that since each RS codeword symbol can be decomposed into ω bits, every ω consecutive pairs of extrinsic probability values $(P_{e,j}^{(1)}(0), P_{e,j}^{(1)}(1))$ are grouped together to represent an RS codeword symbol during the deinterleaving process.

[1] State 1 denotes the all-zero state of the inner encoder.

B. SISO Decoding of the Outer Code

By reading out each row of the deinterleaver and mapping

$$P_e^{(1)} \mapsto P_a^{(2)}, \quad (18)$$

we can obtain the *a priori* probability for each RS coded bit c_j. For simplicity, we will now describe the SISO decoding of an RS codeword and hence drop the codeword index γ. Let

$$P_{a,j}^{(2)}(\theta) = \Pr[c_j = \theta] \quad (19)$$

denote the *a priori* probability of c_j, where $j = 1, 2, \ldots, n\omega$. Its *a priori* log-likelihood ratio (LLR) value is

$$L_{a,j} = \ln\Big(\frac{P_{a,j}^{(2)}(0)}{P_{a,j}^{(2)}(1)}\Big). \quad (20)$$

The *a priori* LLR vector of an RS codeword can be formed

$$\overline{L}_a = [L_{a,1}\ L_{a,2}\ \cdots\ L_{a,(n-k)\omega}\ \cdots\ L_{a,n\omega}]. \quad (21)$$

The ABP algorithm will first sort the *a priori* LLR values based on their magnitudes $|L_{a,j}|$, yielding a refreshed bit indices sequence $\delta_1, \delta_2, \ldots, \delta_{(n-k)\omega}, \ldots, \delta_{n\omega}$, which implies $|L_{a,\delta_1}| < |L_{a,\delta_2}| < \cdots < |L_{a,\delta_{(n-k)\omega}}| < \cdots < |L_{a,\delta_{n\omega}}|$. Since a higher magnitude implies that the bit is more reliable, we know that $c_{\delta_1}, c_{\delta_2}, \ldots, c_{\delta_{(n-k)\omega}}$ are the $(n-k)\omega$ least reliable bits. Let $UR = \{\delta_1, \delta_2, \ldots, \delta_{(n-k)\omega}\}$ denote the set of the unreliable bit indices and $|UR| = (n-k)\omega$, and its complementary set UR^c collects the $k\omega$ reliable bit indices. Based on the set UR, the sorted *a priori* LLR vector is

$$\overline{L}_a^{UR} = [L_{a,\delta_1}\ L_{a,\delta_2}\ \cdots\ L_{a,\delta_{(n-k)\omega}}\ \cdots\ L_{a,\delta_{n\omega}}]. \quad (22)$$

The ABP algorithm will then perform Gaussian elimination on matrix $\mathbf{H}_b$, reducing the columns that correspond to the unreliable bits to weight-1 columns. Let Υ_δ denote the weight-1 column vector with 1 at its δth entry and 0 elsewhere. For matrix $\mathbf{H}_b$, Gaussian elimination reduces column δ_1 to Υ_1, then reduces column δ_2 to Υ_2, and etc. In the end, Gaussian elimination reduces the first $(n-k)\omega$ independent columns to weight-1 columns, resulting in an adapted parity-check matrix $\mathbf{H}'_b$. By reducing the columns w.r.t. the unreliable bits into weight-1 columns, the propagation of the unreliable information during the BP decoding can be prevented [10], making matrix $\mathbf{H}'_b$ more suitable for the iterative BP process.

Let $h_{ij} \in \{0, 1\}$ denote the entry of matrix $\mathbf{H}'_b$ and

$$\mathbf{I}(j) = \{i \mid h_{ij} = 1,\ \forall\ 1 \le i \le (n-k)\omega\}, \quad (23)$$

$$\mathbf{J}(i) = \{j \mid h_{ij} = 1,\ \forall\ 1 \le j \le n\omega\}. \quad (24)$$

The iterative BP process is performed based on the Tanner graph that is associated with matrix $\mathbf{H}'_b$, yielding the extrinsic LLR value for each RS coded bit by

$$L_{e,j} = \sum_{i\in\mathbf{I}(j)} 2\tanh^{-1}\Big(\prod_{\tau\in\mathbf{J}(i)\setminus j} \tanh\Big(\frac{L_{a,\tau}}{2}\Big)\Big). \quad (25)$$

After a moderate number of BP iterations, the *a posteriori* LLR value of each RS coded bit is determined by

$$L_{p,j} = L_{a,j} + \eta L_{e,j}, \quad (26)$$

where $\eta \in (0, 1]$ is the damping factor [11]. Therefore, the *a posteriori* LLR vector of an RS codeword can be formed as

$$\overline{L}_p = [L_{p,1}\ L_{p,2}\ \ldots\ L_{p,(n-k)\omega}\ \ldots\ L_{p,n\omega}]. \quad (27)$$

Note that in order to reduce the decoding complexity, the extrinsic LLR calculation rule of (25) can be simplified to

$$L_{e,j} = \sum_{i\in\mathbf{I}(j)} \Big(\prod_{\tau\in\mathbf{J}(i)\backslash j} \mathrm{sign}(L_{a,\tau}) \cdot \min_{\tau\in\mathbf{J}(i)\backslash j} \{|L_{a,\tau}|\}\Big). \quad (28)$$

Given a random variable ψ, $\mathrm{sign}(\psi) = 0$ if $\psi \geq 0$, or $\mathrm{sign}(\psi) = 1$ otherwise.

It is important to point out that the ABP algorithm itself is also an iterative process. That says if there are multiple Gaussian eliminations, the *a posteriori* LLR vector will be mapped back to the *a priori* LLR vector by

$$\overline{L}_p \mapsto \overline{L}_a. \quad (29)$$

Based on the updated $\overline{L}_a$ vector, the next round bit reliability sorting and Gaussian elimination will be performed. Based on each adapted matrix $\mathbf{H}'_b$, a number of BP iterations will be carried out, delivering both the extrinsic and the *a posteriori* LLR values. The extrinsic probabilities and the *a posteriori* probabilities of the RS coded bits can be determined by

$$P^{(2)}_{e,j}(0) = \frac{1}{1+e^{-L_{e,j}}},\ P^{(2)}_{e,j}(1) = \frac{1}{1+e^{L_{e,j}}}, \quad (30)$$

$$P^{(2)}_{p,j}(0) = \frac{1}{1+e^{-L_{p,j}}},\ P^{(2)}_{p,j}(1) = \frac{1}{1+e^{L_{p,j}}}. \quad (31)$$

With the knowledge of the *a posteriori* probabilities, the reliability matrix $\mathbf{\Pi} \in \mathbb{R}^{q\times n}$ w.r.t. an RS codeword $\overline{C}$ can be obtained. Its entry $\pi_{\mu\nu}$ is the *a posteriori* probability of an RS codeword symbol C_ν being the field symbol ρ_μ and

$$\pi_{\mu\nu} = \Pr[C_\nu = \rho_\mu \mid \overline{\mathcal{Y}}], \quad (32)$$

where $\mu = 1, 2, \ldots, q$ and $\nu = 1, 2, \ldots, n$. Let Ξ_μ denote the binary representation of the field symbol ρ_μ and

$$\Xi_\mu = [\theta_1\theta_2\cdots\theta_\omega \mid \rho_\mu = \sum_{\kappa=1}^{\omega} \theta_\kappa \alpha^{\omega-\kappa} \text{ and } \theta_\kappa \in \{0,1\}]. \quad (33)$$

Reliability $\pi_{\mu\nu}$ can be determined by

$$\pi_{\mu\nu} = \prod_{\kappa=1,\theta_\kappa\in\Xi_\mu}^{\omega} P^{(2)}_{p,(\nu-1)\omega+\kappa}(\theta_\kappa). \quad (34)$$

Every ω consecutive pairs of $(P^{(2)}_{p,j}(0), P^{(2)}_{p,j}(1))$ values will be multiplied in q different permutations, yielding a column of matrix $\mathbf{\Pi}$. It will then be transformed into a multiplicity matrix $\mathbf{M} \in \mathbb{N}^{q\times n}$ with entries $m_{\mu\nu}$. Interpolation will then be carried out based on the instruction of $\mathbf{M}$, yielding an interpolated polynomial $Q \in \mathbb{F}_q[x, y]$ [5]. Finally, factorization will be carried out, finding the y-roots of Q [12]. The coefficients of a y-root form a decoded message candidate.

If the transmitted message is included in the KV decoding output list $\mathcal{L}$, it is considered the KV decoding is successful. Consequently, the deterministic probabilities of the corresponding RS coded bits can be obtained. Let $\hat{c}_j \in \{0,1\}$ denote the decoded RS bit, where $j = 1, 2, \ldots, n\omega$. The deterministic probability of bit $\hat{c}_j$ is

$$\begin{cases} \tilde{P}_j(0) = 1,\ \tilde{P}_j(1) = 0,\ \text{if } \hat{c}_j = 0; \\ \tilde{P}_j(0) = 0,\ \tilde{P}_j(1) = 1,\ \text{if } \hat{c}_j = 1. \end{cases} \quad (35)$$

The deterministic probabilities will be left unchanged in the rest of the iterations and those decoded RS bits will not be decoded again. If the transmitted message is not included in $\mathcal{L}$, the current RS decoding attempt is unsuccessful. Consequently, the system will give the extrinsic probabilities of (30) as the feedback. Hence, after the SISO decoding of all the D RS codewords, the extrinsic probabilities $P^{(2)}_{e,j}$ of the undecoded bits and the deterministic probabilities $\tilde{P}_j$ of the decoded bits will be fed back. They are then interleaved and mapped to the *a priori* probabilities of the interleaved RS coded bits c'_j by

$$P^{(2)}_e/\tilde{P} \mapsto P^{(1)}_a. \quad (36)$$

With the assistance of the deterministic probabilities, the next round MAP decoding is functioning with a portion of known *a priori* information. It improves the SISO decoding of the inner code. Note that the identities of the decoded bits will be memorized by the system and this can be realized by utilizing a binary indicator of size $Dn\omega$. Once the next round MAP decoding is finished, only the extrinsic probabilities of the undecoded bits will be determined by (17). The iterative soft decoding terminates when either all the D RS codewords have been decoded or the maximal iteration number N_{ITER} is reached.

Note that since the deterministic probabilities will become the known *a priori* information for the next round MAP decoding, having an accurate KV decoding output validation is important. In practice, this can be realized by using the cyclic redundant check (CRC) code.

IV. Performance Evaluations And Discussions

The performance of the proposed decoding algorithm is evaluated by measuring RSCC code's bit error rate (BER) performance over the additive white Gaussian noise (AWGN) channel. The binary phase shift keying (BPSK) modulation is used. In SISO decoding of the outer code, there are two parity-check matrix adaptations and two BP iterations after each adaptation. After an extensive search, such an ABP decoding setup prevails the alternatives in performance. The KV decoding is parameterized by its designed factorization output list size l and $l = \max\{|\mathcal{L}|\}$. In our evaluations, it is set $l = 10$. The depth of the block interleaver is 10. The proposed algorithm is compared with the Viterbi-BM, the MAP-KV and the MAP-ABP-KV algorithms. Note that the MAP-ABP-KV algorithm corresponds to the proposed algorithm without any iteration. Decoding gains are quantized at the BER of 10^{-5}.

Fig.3 shows the iterative decoding performance of the (63, 55) RS – 16-state NSC concatenated code. The generator polynomials of the inner code are $G_1(x) = 1 + x^2 + x^3$ and $G_2(x) = 1+x^2+x^3+x^4$. It shows the iterative soft decoding achieves significant performance gains over the benchmark algorithms. For example, iterative decoding with 10 iterations achieves 1.1dB, 0.85dB and 0.55dB performance gains over the Viterbi-BM, the MAP-KV and the MAP-ABP-KV algorithms, respectively. By increasing the number of iterations, larger performance gains can be achieved. However, it should

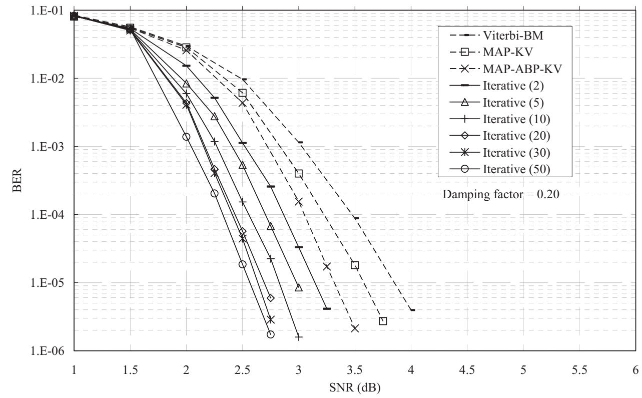

Fig. 3. Iterative soft decoding performance of the (63, 55) RS – 16-state NSC concatenated code.

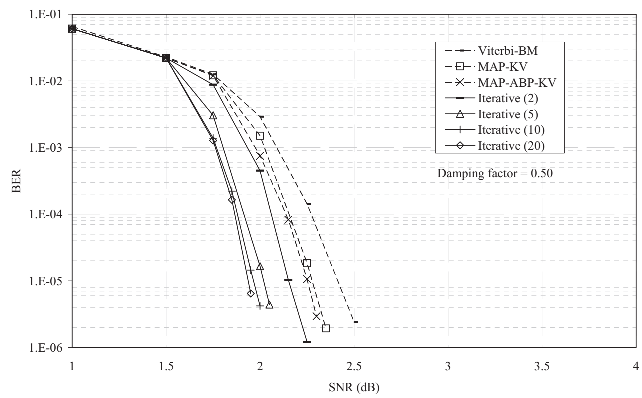

Fig. 4. Iterative soft decoding performance of the (255, 239) RS – 64-state NSC concatenated code.

be pointed out that the decoding gains are realized at the expense of computational complexity. During each iteration, the proposed algorithm requires at most $O(12\Omega Dn\omega + 4D(n\omega)^2)$ floating point operations, $O(2Dn\omega(n\omega - k\omega)^2)$ binary operations and $O(2D(k-1)^2 l^5)$ finite filed arithmetic operations, respectively. The computational cost will be scaled up linearly w.r.t. to the iteration number N_{ITER}.

The (255, 239) RS – 64-state NSC concatenated code with the inner code generator polynomials $G_1(x) = 1+x+x^3+x^4+x^6$ and $G_2(x) = 1+x^3+x^4+x^5+x^6$ is employed in various wireless communication systems. Fig.4 shows the iterative soft decoding performance of this code. Note that the extrinsic LLR calculation of (28) is utilized to decode the outer code. Again, it shows significant performance improvements can be made. Performing iterative soft decoding with 10 iterations achieves 0.45dB gain over the Viterbi-BM algorithm. However, it also turns out that marginal performance improvement can made with an iteration number greater than 10.

Finally, it can be aware that the SISO decoding of the outer code is suboptimal. Its decoding capability can be further strengthened by improving either the ABP or the KV process. Specifically, the ABP process can be improved by further restructuring the sorted LLR vector to allow bits of UR^c being corrected by the BP decoding [10], and the KV process can be improved by increasing l [5]. However, our investigation shows those approaches offer marginal performance improvement for the concatenated codes. Considering the presentation limit, they are not shown in this paper.

V. Conclusion

An iterative soft-decision decoding algorithm has been proposed for the popular RSCC codes. The MAP algorithm and the ABP-KV algorithm are used to decode the inner and the outer codes, respectively. The ABP algorithm is capable to deliver the extrinsic and the *a posteriori* probabilities of the RS coded bits with a polynomial-time complexity. It enables the extrinsic probabilities of the RS coded bits to be iterated in a soft information exchange decoding mechanism. Consequently, the concatenated code's error-correction potential can be well exploited. Our simulation results show that significant performance gains can be achieved over the conventional decoding algorithms. Therefore, the proposed algorithm can be considered for upgrading the existing communication systems in which the RSCC code is employed.

Acknowledgement

This research is supported by the National Basic Research Program of China (973 Program) with project ID 2012CB316100 and the National Natural Science Foundation of China (NSFC) with project ID 61001094.

References

[1] G. D. Forney, *Concatenated Codes.* MIT Press, Cambridge, MA, 1966.
[2] R. D. Cideciyan, E. Eleftheriou and M. Rupf "Concatenated Reed-Solomon/convolutional coding for data transmission in CDMA-based cellular systems," *IEEE Trans. Commun.*, vol. 45 (10), pp. 1291-1303, Oct. 1997.
[3] Consultative Committee for Space Data Systems, "Recommendation for space data systems standard, telemetry channel coding," CCSDS 101.0-B-6, *Blue Book*, issue 6, Oct. 2002.
[4] E. Paaske, "Improved decoding for a concatenated coding system recommended by CCSDS," *IEEE Trans. Commun.*, vol. 38 (8), pp. 1138-1144, Aug. 1990.
[5] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 49 (11), pp. 2809-2825, Nov. 2003.
[6] J. Zhang, J. You, L. Lu and X. Ma, "An iterative soft-decision decoding algorithm for conventional concatenated codes," in *Proc. the 4th Int. Conf. Commun. and Netw. in China*, Xi'an, China, Aug. 2009.
[7] G. Schmidt, V. Sidorenko and M. Bossert, "Collaborative decoding of intereleaved Reed-Solomon codes and concatenated code designs," *IEEE Trans. Inform. Theory*, vol. 55 (7), pp. 2991-3012, Jul. 2009.
[8] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes (1)," in *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May. 1993.
[9] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol.20 (2), pp. 284-287, Mar. 1974.
[10] J. Jiang and K. Narayanan, "Iterative soft-input-soft-output decoding of Reed-Solomon codes by adapting the parity check matrix," *IEEE Trans. Inform. Theory*, vol. 52 (8), pp. 3746-3756, Aug. 2006.
[11] M. El-Khamy and R. McEliece, "Iterative algebraic soft-decision list decoding of Reed-Solomon codes," *IEEE J. Sel. Area in Commun.*, vol.24 (3), pp. 481-490, Mar. 2006.
[12] L. Chen, R. A. Carrasco, M. Johnston and E. G. Chester, "Efficient factorisation algorithm for lsit decoding algebraic-geometric and Reed-Solomon codes," *Proc. IEEE Int. Conf. Commun.*, pp. 851-856, Glasgow, UK, 2007.